

M2 MPRI - Programmation Probabiliste - μ -PPL

Guillaume Baudart et Christine Tasson

1 Premiers exemples.

Implémenter les exemples suivant et vérifier expérimentalement les résultats de différents algorithmes d'inférence.

Exercice 1. *Temps d'arrêt*

On compte le nombre de lancers d'une pièce équilibrée avant que la pièce retombe sur face.. Quelle est la distribution associée ?

Exercice 2. *Pièce biaisée ou équilibrée*

Utiliser le conditionnement par rejet pour modéliser une pièce équilibrée en utilisant deux lancers d'une pièce biaisée.

Exercice 3. *Coin*

Quelle est la distribution modélisée par le programme suivant.

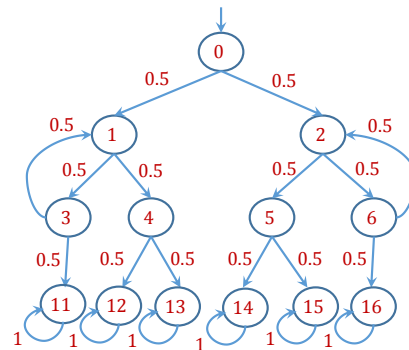
```
def coin(obs: list[int]) → float:
    p = sample(Uniform(0, 1))
    for o in obs:
        observe(Bernoulli(p), o)
    return p

with ImportanceSampling(num_particles=10000):
    dist: Categorical = infer(coin, [0, 0, 0, 0, 0, 0, 0, 0, 1, 1])
    print(dist.stats())
    viz(dist)
```

2 Le dé de Knuth-Yao

Knuth et Yao ont proposé un algorithme qui permet de simuler un dé parfait à 6 faces à partir d'une pièce équilibrée [1]. L'algorithme correspond à la chaîne de Markov discrete représentée ci-dessous [3].

À partir de l'état initial, on suit les transitions en fonction du résultat d'un lancé de pièce (probabilité 0.5 sur chacune des deux branches). On renvoie la valeur obtenue dès qu'on arrive sur une feuille. Les 6 feuilles numérotées de 11 à 16 correspondent aux 6 faces du dé.



Exercice 4.

Implémenter ce modèle et vérifier à l'aide de plusieurs méthodes d'inférence qu'on obtient bien un dé équilibré.

3 La régression linéaire de Galton

En 1886, Francis Galton mesure le taux de régression de la taille entre les parents et leurs enfants [4].

Il utilise pour cela la table des fréquences *Nombre d'enfants Adultes de tailles variées, nés de 205 parents de tailles variées*. Il divise les données en plusieurs sousgroupes en fonction de la taille moyenne des deux parents. Il calcule alors la médiane des tailles de leurs enfants et la compare à la taille médiane des parents. Il reconnaît une ligne droite.

Exercice 5.

Proposer un programme probabiliste qui permette de modéliser la régression linéaire et de reproduire le résultat de Galton, en trouvant la droite qui corresponde au mieux avec les données.

4 TrueSkillsTM

TrueSkillsTM est un système de classement développé par Microsoft pour Xbox Live.¹

Le niveau de chaque joueur est représenté par une distribution gaussienne. On suppose initialement que pour tous les joueurs P , $skill_P \sim \mathcal{N}(100, 10)$.

À chaque duel, la performance de chacun des joueurs suit une distribution gaussienne centrée sur leur niveau avec une variance fixe. On observe alors que la performance du vainqueur W est supérieure à celle du vaincu L ce qui mets à jours les variables aléatoires $skill_W$ et $skill_L$.

$$perf_W \sim \mathcal{N}(skill_W, 15)$$

$$perf_L \sim \mathcal{N}(skill_L, 15)$$

$$perf_W > perf_L$$

Exercice 6.

Implémenter ce modèle pour trois joueurs A, B et C, et trois duels :

- A gagne contre B,
- B gagne contre C,
- A gagne contre C.

Vérifier que les résultats obtenus correspondent à ce qu'on attend.

Exercice 7.

Implémenter un modèle plus général qui prend en paramètre un tableau de joueur avec leur niveau initial et une liste de duels. On pourra représenter un duel par une pair d'entiers (i, j) où i et j correspondent aux indices des joueurs dans le tableau. Par convention, le vainqueur du duel sera toujours i .

Références

- [1] D. Knuth et A. Yao. *Algorithms and Complexity : New Directions and Recent Results, chapter The complexity of nonuniform random number generation*. Academic Press, 1976.
- [2] R. Herbrich, T. Minka et T. Graepel *TrueSkillTM : A Bayesian Skill Rating System*. NIPS, 2006.
- [3] A. Gordon, T. Henzinger, A. Nori et S. Rajamani *Probabilistic programming*. FOSE, 2014.
- [4] Galton *Regression Towards Mediocrity in Hereditary Stature*, Journal of the Anthropological Institute of Great Britain and Ireland, 15, 246-263, 1886.

1. <https://en.wikipedia.org/wiki/TrueSkill>